

# KXChang



K r y p t o M a g i k

# KXChang

## Key Exchange Algorithm

# KXChang

- Invented in 2019
- Reference implementation in Python

# Design Goals

- To be resistant to DLP attacks
- To have smaller key lengths than traditional DiffieHellman

# Key Generation

- Generate 2  $N$  bit integers, let them be  $A$ ,  $B$  and let them not be equal
- Establish a modulus  $N$  as the product of  $A$  and  $B$
- Choose two numbers  $X$  and  $Y$  between 1 and  $A - 1$
- Let  $X$  be the public key

# Key Generation

- Take  $X + Y$  modulo  $N$  and call that  $Z$ . Let  $Z$  be the secret key

# Key Exchange Setup

- Alice and Bob generate SK, PK, N
- Either Bob or Alice sends their public key over
- Alice and Bob both send their domain modulus over

# Key Exchange Phase 1

- Alice and Bob have chosen to use Bob's public key. Alice and Bob both raise Bob's public key to their secret exponent modulo the shared modulus. They exchange phase 1.



# Key Exchange Phase 2

- Alice and Bob compute phase1 raised to their secret exponent modulo  $S$  arriving and the secret modulus.

# Key Exchange Phase 3

- Alice and Bob elect Alice to select a number  $Y$  between 1 and  $S - 1$
- Alice sends the number to Bob
- Alice and Bob compute phase 3 by raising  $Y$  to their secret powers modulo the secret modulus

# Key Exchange Phase 4

- Alice and Bob raise phase3 to their secret exponent modulo the secret modulus arriving at the shared secret.

# Cryptanalysis

- One can use the discrete logarithm to compute the shared modulus
- The shared secret cannot be easily computed based on the secret key generated from the DL